

March 14th-18th, 2016

ENGINEERING DECISIONS BEHIND

WORLD OF TANKS

GAME CLUSTER

Maksim Baryshnikov, Wargaming

#GDC2016


Minsk
Norwegian Sea
Murmansk
Naryan-Mar
Vorkuta
Norilsk
Novy Urengoy
ICELAND
Reykjavik
Arkhangelsk
RUSSIA
FINLAND
Petrozavodsk
Syktyvkar
Surgut
Yakutsk
SWEDEN
Kirov
Perm
Tobolsk
UNITED KINGDOM
DENMARK
Moscow
Kazan
Chelyabinsk
Omsk
Novosibirsk
Krasnoyarsk
Bratsk
Abakan
Irkutsk
Chita
Tahe
Cork
London
GERMANY
POLAND
Lipetsk
Samara
Luxembourg
Prague
Lviv
UKRAINE
Kamyshin
Aktobe
Astana
Semipalatinsk
Manzhouli
Nenjiang
FRANCE
AUSTRIA
ROMANIA
Rostov-on-Don
Atyrau
KAZAKHSTAN
Altay
MONGOLIA
Ljubljana
Harbin
Marseille
ITALY
BULGARIA
Nalchik
Aktau
Ürümqi
Abag Banner
SPAIN
GREECE
TURKEY
AZERBAIJAN
UZBEKISTAN
KYRGYZSTAN
Dunhuang
CHINA
Baotou
Beijing
NORTH KOREA
Pyongyang
Lisbon
TAJIKISTAN
Ashgabat
Urmia
Mizhi
Jinan
Algiers
Tunis
Rhodes
Tehran
Hotan
Golmud
Gilgit
Linxia
SOUTH KOREA
Xi'an
Rabat
TUNISIA
LEBANON
IRAQ
IRAN
AFGHANISTAN
Gar
Gyegu
MOROCCO
Benghazi
JORDAN
PAKISTAN
Chengdu
Shanghai
Ghadames
Cairo
KUWAIT
Kerman
Lhasa
ALGERIA
New Delhi
NEPAL
Changsha
LIBYA
EGYPT
SAUDI ARABIA
WESTERN SAHARA
Kufra
Abu Dhabi
Karachi
Kota
BANGLADESH
Kunming
TAIWAN
Naha
INDIA
Wadi Halfa
Jeddah
OMAN
Jiangmen
MAURITANIA
LAOS
Mumbai
MYANMAR
Vigan
ERITREA
YEMEN
SUDAN
VIETNAM



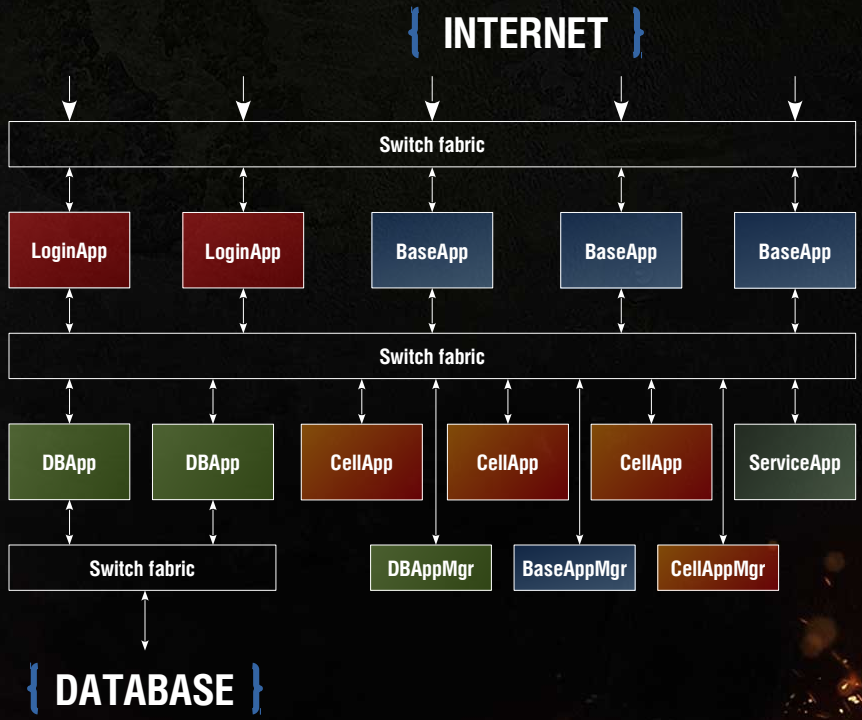
Massively multiplayer online game featuring early to mid-20th century era fighting vehicles.

The focus is on player vs. player gameplay with each player controlling an armored vehicle.


bigWORLD®
TECHNOLOGY


CLUSTER ANATOMY
HOW SINGLE CLUSTER WORKS

CLUSTER COMPONENTS



CLUSTER COMPONENTS

LoginApp processes

Responsible for logging user in. LoginApps have public IP.

CellApp processes

Power actual tank battles. Load is dynamically balanced among CellApps in real-time.

BaseApp processes

Proxy between user and CellApp. Runs all hangar logic. BaseApps have public IP.

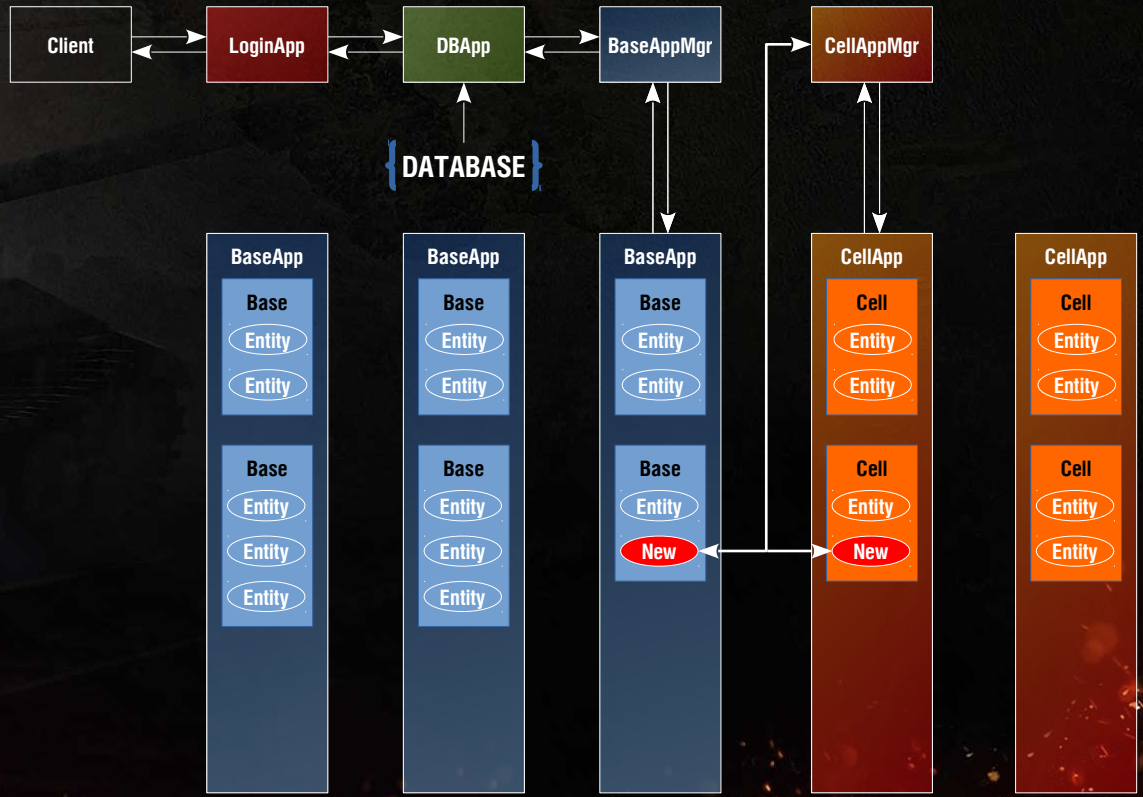
DBApp processes

DBApps persist user data to the database.

*Mgr processes

Manage instances of corresponding *App processes.

CLUSTER COMPONENTS

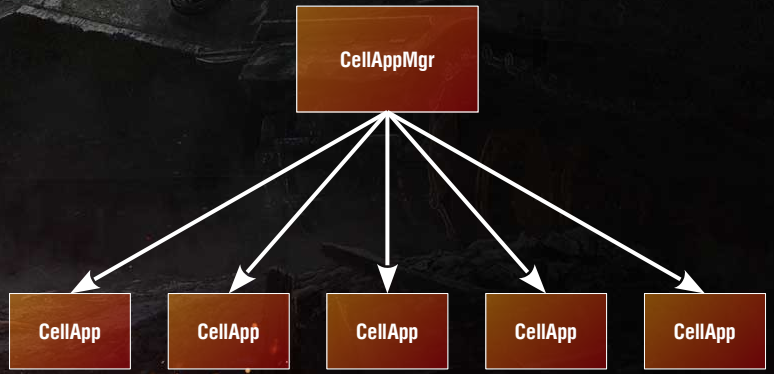


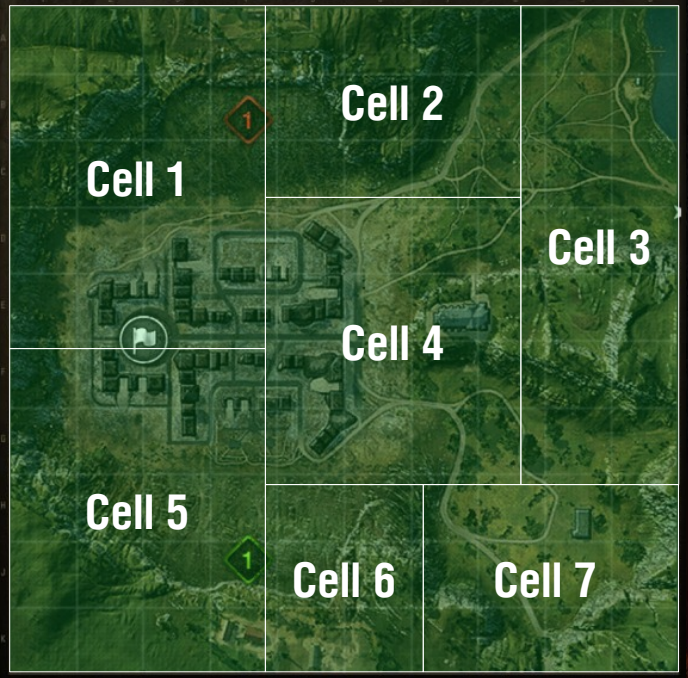

CLUSTER ANATOMY
HOW BATTLE IS HANDLED WITHIN CLUSTER INFRASTRUCTURE

SPACES (BATTLE ARENAS)

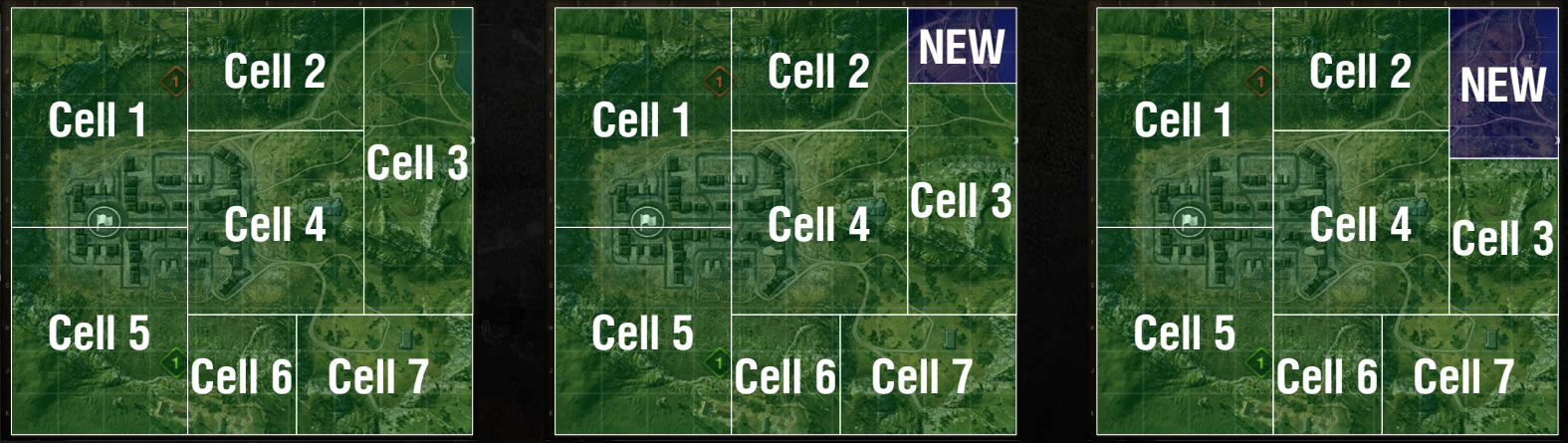
Cell load — amount of time cell spends in calculation of game situation divided by length of game tick.

CellAppMgr changes cells' sizes in real-time in order to keep load of every cell below configured threshold.





MAINTAINING CELL LOAD



CellAppMgr can also add additional cells to space in order to maintain each cell's load below configured value.

AREA OF INTEREST



Rectangular axis-aligned AoI works best because its boundaries are aligned with chunks' and cells' boundaries.

AoI distance is configurable. **500m** initially chosen for the game design reasons.

AoI is not equal to **Area of Visibility**. AoV is circular and fits into AoI. AoI is used by server to optimize AoV checks and keep track of potential interaction between entities.

Visibility is raycast-based.

“GHOST IN THE CELL”

Real Entity is a master instance of an entity.

Ghost Entity is a copy of an entity from a nearby cell. The ghost copy contains all entity data that may be needed.

Each cell communicates with it's neighbors in order to maintain list of it's real entities which have to be ghosted in adjacent cells.

When crossing a cell's border, ghosts turn into real entities.



AREA OF INTEREST



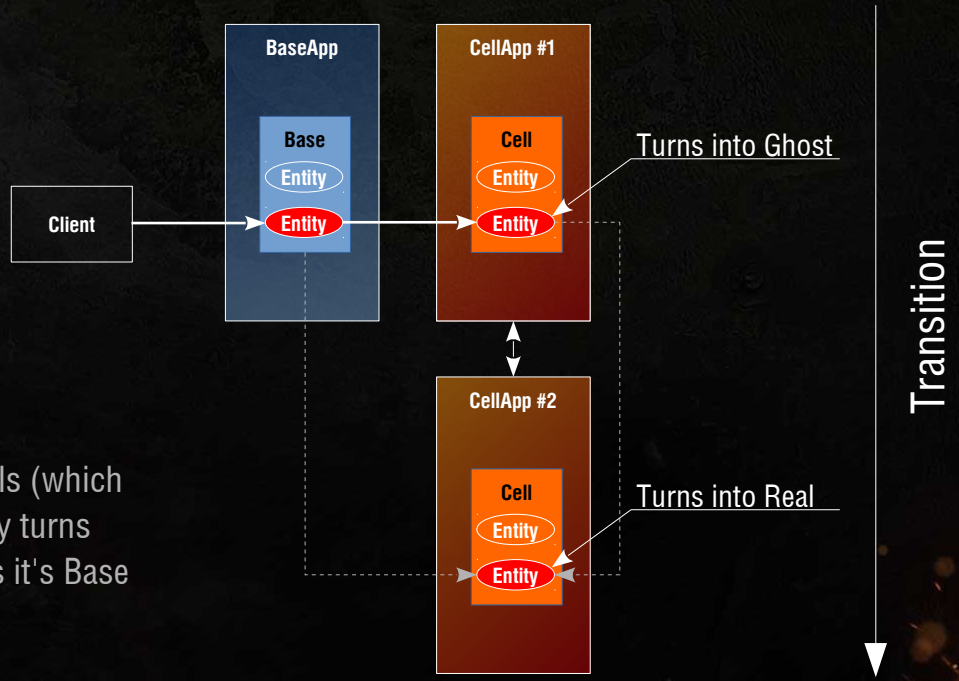
Rectangular axis-aligned AoI works best because its boundaries are aligned with chunks' and cells' boundaries.

AoI distance is configurable. **500m** initially chosen for the game design reasons.

AoI is not equal to **Area of Visibility**. AoV is circular and fits into AoI. AoI is used by server to optimize AoV checks and keep track of potential interaction between entities.

Visibility is raycast-based.

“GHOST IN THE CELL”: CROSSING THE BORDER



While transitioning between two adjacent Cells (which may reside in different CellApps), Ghost entity turns into Real and vice versa. BaseApp reconnects it's Base entity to the new Real.

CROSS-CELL FIRE

Cross-cell fire is a common situation.

Shell is attached to **ProjectileMover** entity, which encapsulates trajectory calculation information.

ProjectileMover crosses cells' borders like other entities.

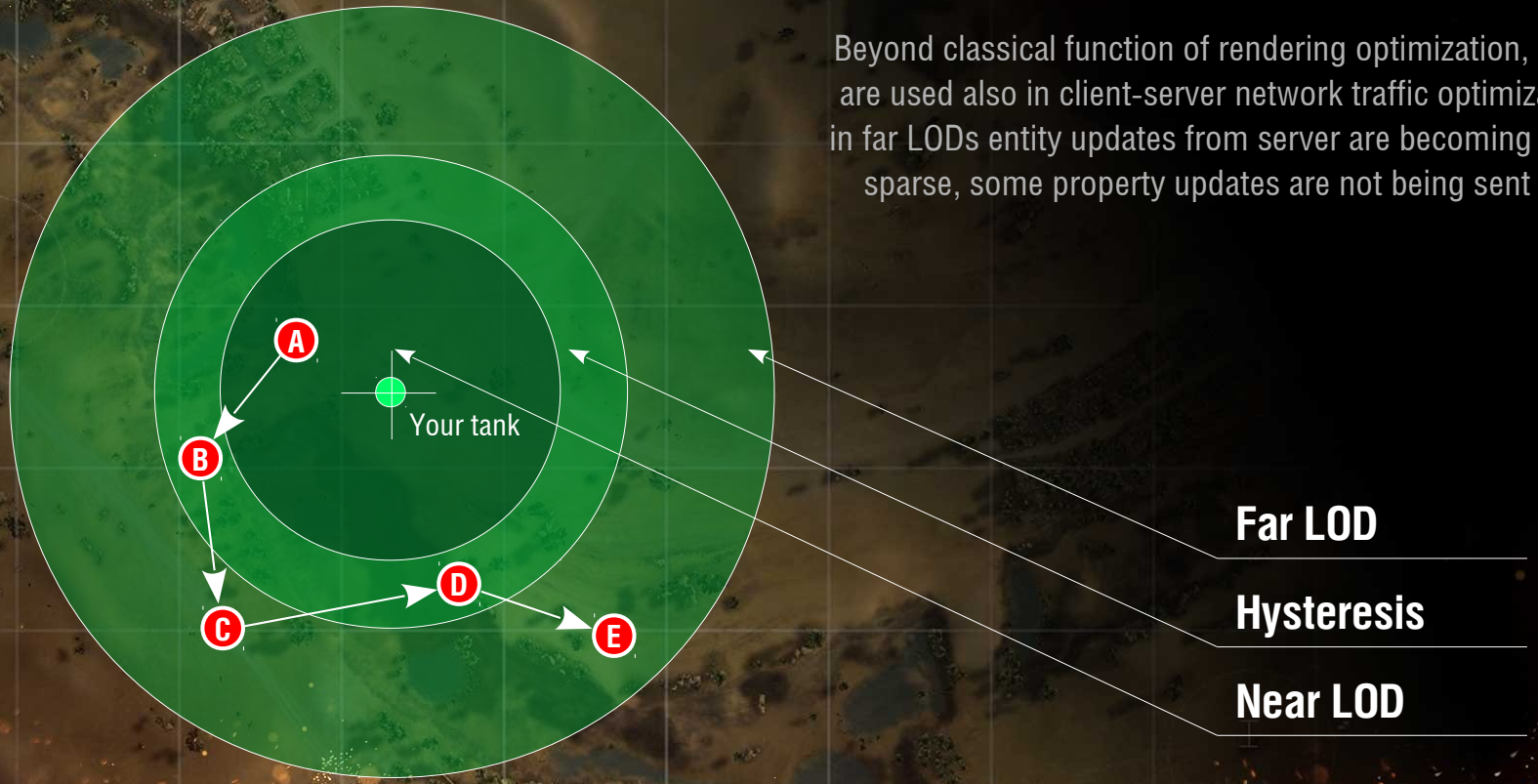
Hit can be checked using Ghost entity.



Due to the fact, that cells can reside in different CellApps, which are spread across physical machines, shell that has been fired from one server can hit the target on another one.

LEVEL OF DETAILS

Beyond classical function of rendering optimization, LODs are used also in client-server network traffic optimization: in far LODs entity updates from server are becoming more sparse, some property updates are not being sent at all.



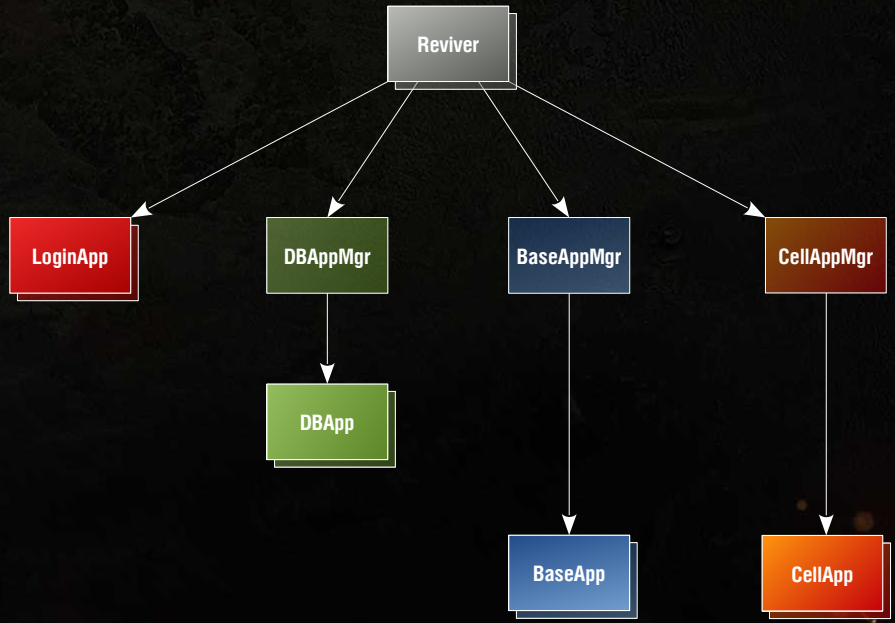

CLUSTER ANATOMY
FAULT TOLERANCE

FAULT TOLERANCE: SENTINELS

Reviver — a watchdog process used to restart other processes that fail.

Reviver processes are typically started on machines reserved for fault tolerance purposes.

***Mgr** processes restart failed ***App** processes



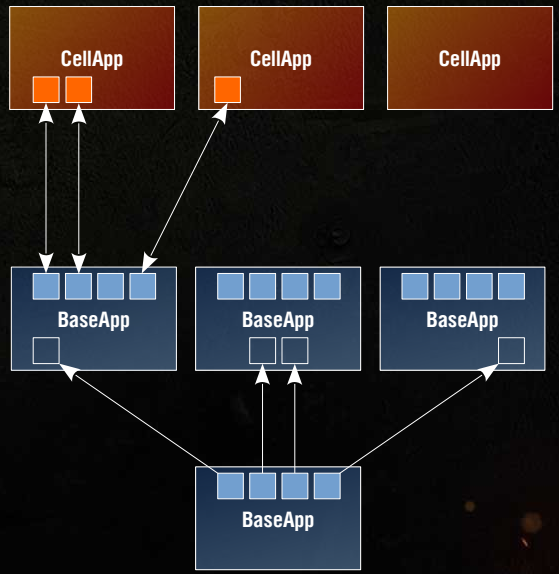
FAULT TOLERANCE: BACKUPS

Entities in **CellApp** store their back-up data in corresponding **BaseApp** entity

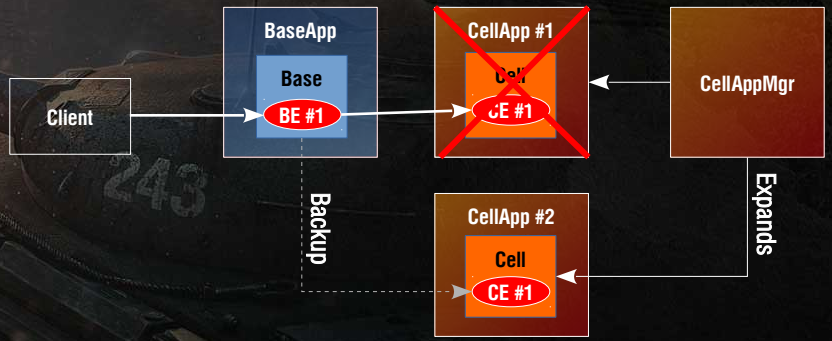
BaseApp backs up its entities to other BaseApps, holds cell entity backup data.

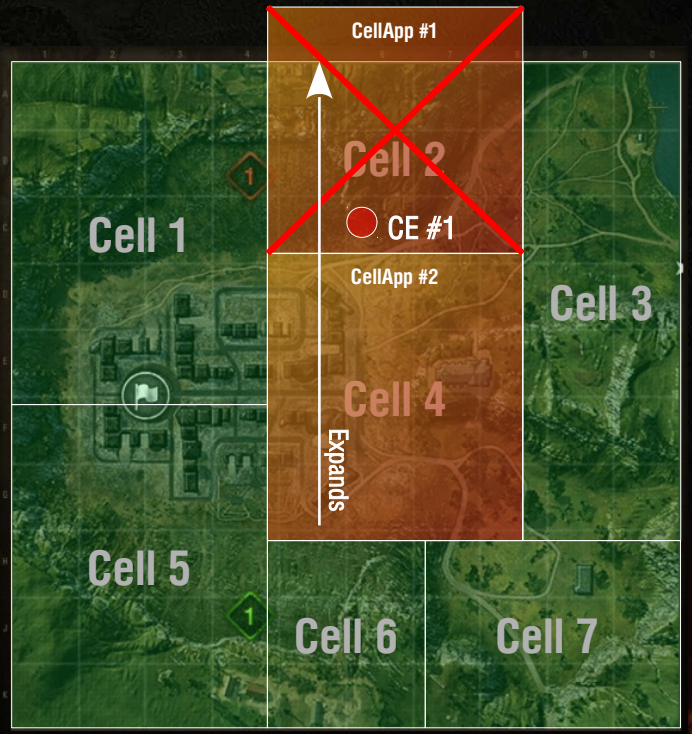
Upon **CellApp** crash, cell entities will be restored from latest backup available.

If a **BaseApp** dies, each of its entities is restored on the **BaseApp** that was backing it up.



FAULT TOLERANCE: CELLAPP DEATH





- CellApp #1 dies so does Cell #2
- CellAppMgr removes Cell #2 and expands Cell #4 to cover former Cell #2 area
- Cell Entity #1 (CE #1) is being restored from backup in Cell #4

From player's perspective this looks like 1-2 seconds lag

CAP THEOREM

Single cluster targets **Availability** and **Partition Tolerance** in terms of CAP theorem

AP approach in this case means that **battle state** in case of components failure is eventually consistent (among server and all connected clients)

*In theoretical computer science, the CAP theorem, also known as Brewer's theorem, states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees: Consistency, Availability and Partition Tolerance



CLIENT & SERVER

PROGRAMMING

PROGRAMMING LANGUAGES

C++

All server components (*Apps, *Mgrs, etc), communication API (Mercury API) as well as CPU-intensive server-side game logic modules. Client core also written in C++.

Python is used for **game logic programming** (both client- and server-side).
Most of *Apps have built-in Python 2.7 interpreter (with disabled garbage collector).




GEOGRAPHICALLY DISTRIBUTED
CLUSTER-OF-CLUSTERS

WORLD OF TANKS: RU



CLUSTER-OF-CLUSTERS



CAP THEOREM

Multi cluster targets **Consistency** and **Partition Tolerance** in terms of CAP theorem

CP approach in this case means that **account state** is consistent among infrastructure components. This sacrifices Availability of the game for a particular client in case of **Periphery cluster** failure or network unavailability

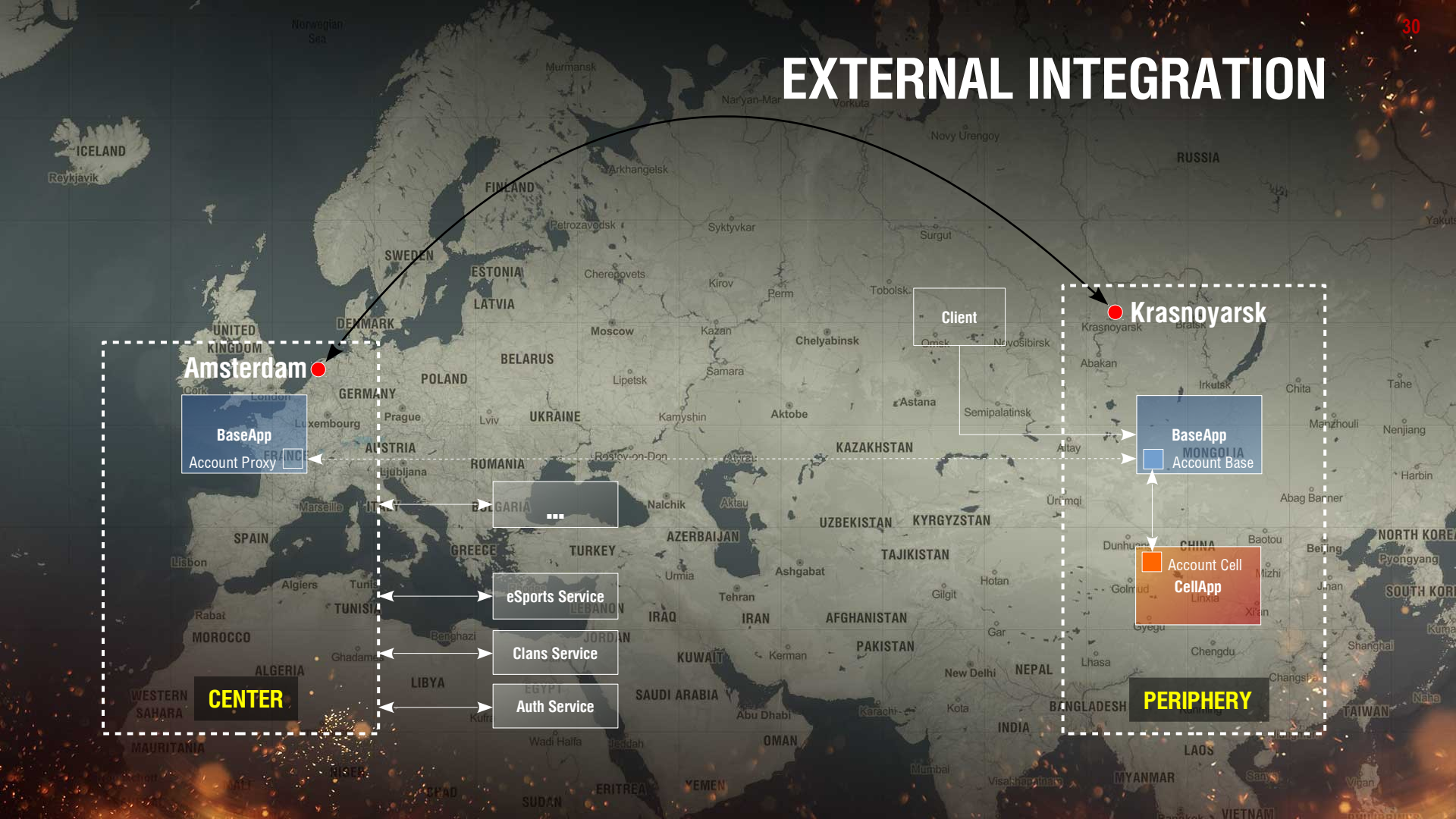
*In theoretical computer science, the CAP theorem, also known as Brewer's theorem, states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees: Consistency, Availability and Partition Tolerance



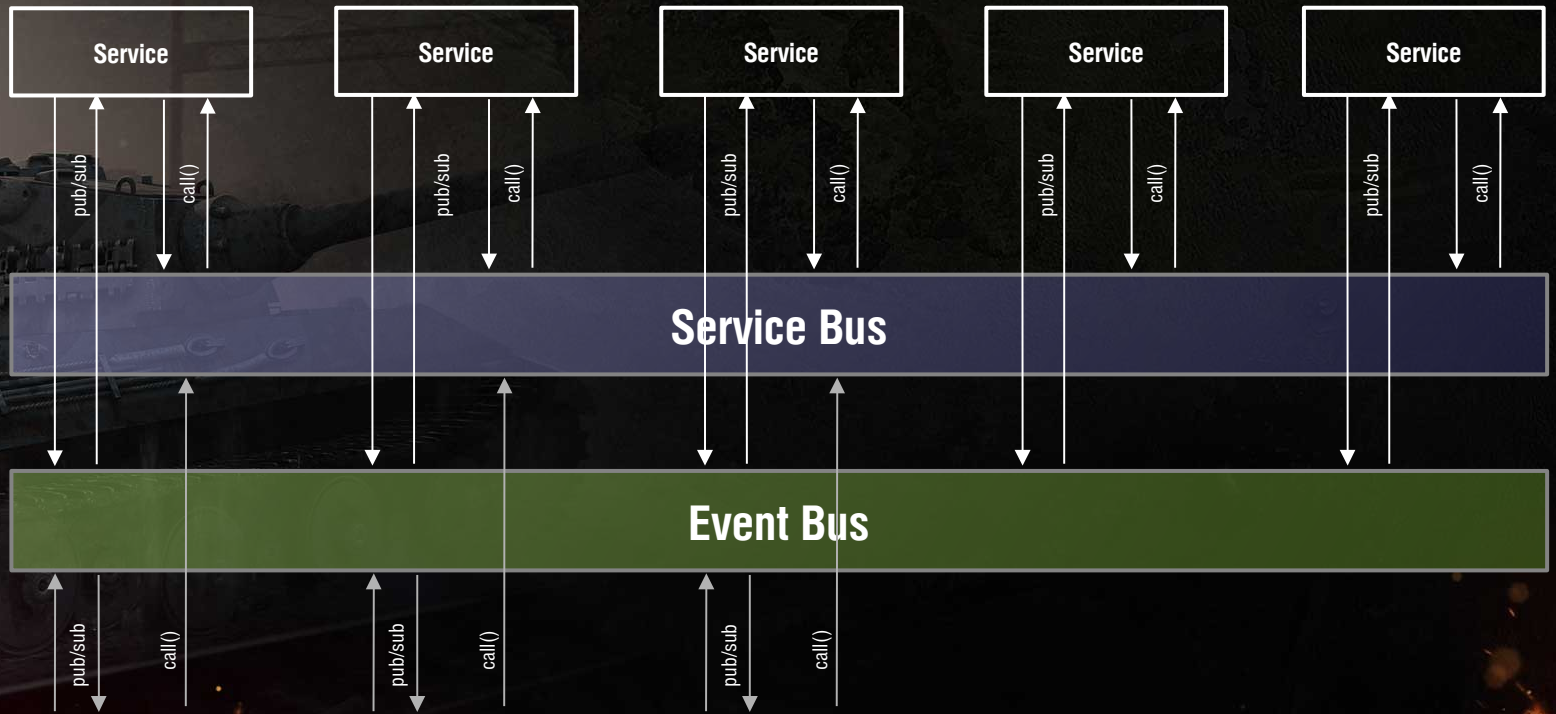
GAME SERVER INFRASTRUCTURE

EXTERNAL INTEGRATION

EXTERNAL INTEGRATION



EXTERNAL INTEGRATION: EVENT DRIVEN SOA

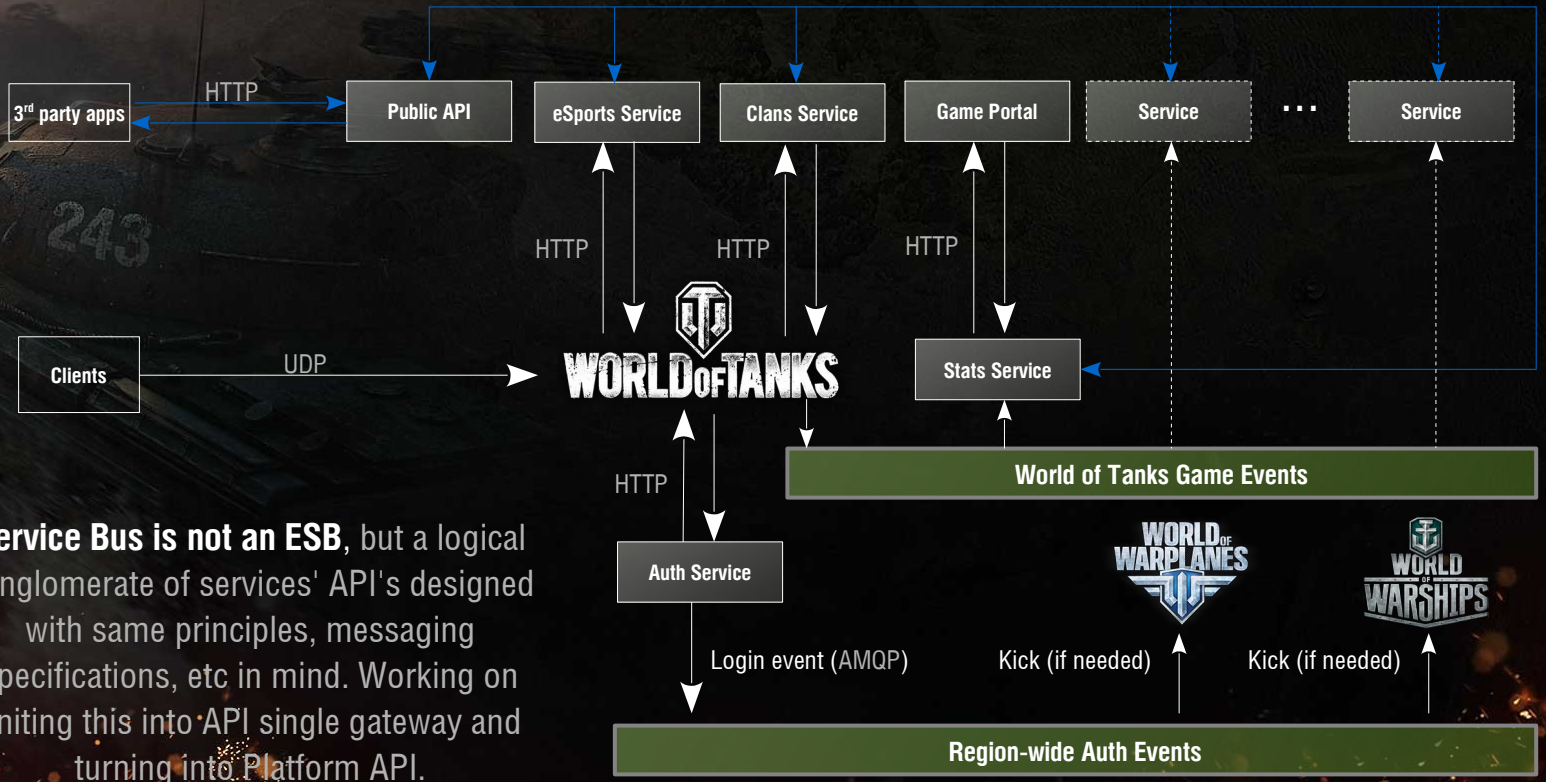


EXTERNAL INTEGRATION: EVENT BUS



Not a single bus, but a composition of buses with different purposes. There are entity state transfer buses, signaling buses, commit logs etc. Some of these buses are region-wide and same across WoT, WoWP, WoWS, some are not.

EXTERNAL INTEGRATION: EXAMPLE



Service Bus is not an ESB, but a logical conglomerate of services' API's designed with same principles, messaging specifications, etc in mind. Working on uniting this into API single gateway and turning into Platform API.

EXTERNAL INTEGRATION: CONSISTENCY

In case of **message loss or link outage** some state information may become stale or even inconsistent.

- State transfers are full (not incremental) → this heals potentially lost state updates in Stats Service (and others) upon arrival of the next state update message.
- All services look up same information in same place, so even stale information **looks consistent.**
- There are watchers in master data storages, which check data consistency and heal if necessary.

Almost any type of information inconsistency will heal automatically.





WORLD OF TANKS IN

NUMBERS

LARGEST MULTI-CLUSTER

30+ million players

Peak of 1.1+ million players simultaneously online

200+ logins/sec, spikes to 1000+

100+ battles started every second

3000+ state exports to external services per second

500+ Gb of accounts data kept in memory

SUMMARY



BigWorld as a core technology

Components designed as separate processes

C++ / Python (built-in interpreter, GC disabled)

Network abstraction / Asynchronous scripting

Service architecture for extended functionality

Message & Service “buses” are important part



DO YOU HAVE ANY

QUESTIONS?

Maxim Baryshnikov

m_baryshnikov@wargaming.net